

L'anatomia di un modello HTML5

Un modello HTML in genere include le seguenti parti:

1. *La dichiarazione del tipo di documento (o doctype)*
2. *Il Tag <html>*
3. *La codifica dei caratteri*
4. *Il meta tag viewport*
5. *<title>, description e author*
6. *Meta tag di OpenGraph per le social card*
7. *Favicon e icone touch*
8. *Collegamenti a fogli di stile e script*

Oltre alla dichiarazione del tipo di documento e il tag <html>, gli elementi sopra elencati si troveranno principalmente all'interno del tag <head> del modello HTML.

Il tipo di documento HTML5

Il tuo modello HTML5 deve iniziare con una dichiarazione del tipo di documento o *doctype*. Un *doctype* è semplicemente un modo per dire al browser che tipo di documento sta guardando. Nel caso dei file HTML, indica la versione dell'HTML. Il *doctype* dovrebbe essere sempre il primo elemento in cima a qualsiasi file HTML. Molti anni fa, la dichiarazione *doctype* era un brutto pasticcio difficile da ricordare, spesso specificato come "XHTML Strict" o "HTML Transitional".

Con l'avvento di HTML5, quei brutti tag indecifrabili sono spariti e ora tutto ciò di cui hai bisogno è questo:

```
<!doctype html>
```

Il *doctype* può essere scritto in maiuscolo, minuscolo o misto. Noterai che il "5" è vistosamente mancante nella dichiarazione. Sebbene l'attuale iterazione del markup web sia nota come "HTML5", in realtà è solo un'evoluzione dei precedenti standard HTML e le specifiche future saranno semplicemente uno sviluppo di ciò che abbiamo oggi.

Il tag <html>

Dopo il *doctype* in qualsiasi documento HTML c'è il tag <html> :

```
<html lang="en">
```

Questo non ha subito alcun cambiamento significativo dall'avvento di HTML5. Nello snippet di codice sopra, abbiamo incluso l'attributo *lang* con un valore di "en", che specifica che il documento è in inglese. Questo non è necessario per la convalida di una pagina, ma riceverai un avviso dalla maggior parte dei validatori se non lo includi.

L' <html> è diviso in due parti, le sezioni <head> e <body>. La <head> contiene informazioni importanti sul documento che non vengono visualizzate all'utente finale, come la codifica dei caratteri e i collegamenti ai file CSS e possibilmente JavaScript. Il <body> contiene tutto ciò che viene visualizzato nel browser: testo, immagini e così via.

Codifica dei caratteri del documento HTML

La prima riga all'interno della <head> di un documento HTML è quella che definisce la codifica dei caratteri per il documento stesso. Questa è una funzionalità facoltativa e non causerà alcun avviso nei validatori, ma è consigliata per la maggior parte delle pagine HTML:

```
<meta charset="utf-8">
```

In quasi tutti i casi, utf-8 è il valore che utilizzerai nei tuoi documenti.

Nota: per garantire che alcuni browser meno recenti leggano correttamente la codifica dei caratteri, l'intera dichiarazione di codifica dei caratteri deve essere inclusa da qualche parte entro i primi 512 caratteri del documento. Dovrebbe anche apparire prima di qualsiasi elemento basato sul contenuto (come il <title> di cui parleremo più avanti).

Il meta elemento della finestra

Il meta elemento viewport è una funzionalità che vedrai in quasi tutti i modelli HTML5. È importante per il web design reattivo e il design mobile-first:

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

Questo <meta> include due attributi che funzionano come un insieme di nomi/valori. In questo caso, il name è impostato su *viewport* e il valore è *width=device-width, initial-scale=1*. Viene utilizzato solo da dispositivi mobili. Noterai che il valore ha due parti, descritte qui:

- *width=device-width*: la larghezza in pixel dell'area visibile in cui si desidera visualizzare il sito Web.
- *initial-scale*: deve essere un numero positivo compreso tra 0.0 e 10.0. Un valore di "1" indica che esiste un rapporto 1:1 tra la larghezza del dispositivo e le dimensioni della finestra.

Puoi leggere un po' di più su queste funzionalità dei meta tag in rete, ma per ora sappi che, nella maggior parte dei casi, questo elemento meta con queste impostazioni è il migliore per i siti Web reattivi e mobile-first.

Il tag <title>, <description>, <author>

La sezione successiva del modello HTML contiene le tre righe seguenti:

```
<title>A Basic HTML5 Template</title>  
<meta name="description" content="A simple HTML5 Template for new projects.">  
<meta name="author" content="SitePoint">
```

Questi elementi fanno parte dell'HTML da molto tempo, quindi non c'è niente di troppo nuovo qui. Il <title> è ciò che viene visualizzato nella barra del titolo del browser (ad esempio, quando si passa sopra una scheda del browser). Questo elemento è l'unico elemento obbligatorio all'interno del <head>.

Gli altri due sono <meta> facoltativi che definiscono una descrizione per scopi SEO.

Open Graph Meta Tag per i social media

Come accennato, tutti i meta elementi sono opzionali ma molti hanno vantaggi per il SEO e il marketing sui social media. La sezione successiva del nostro template HTML5 include alcune di queste opzioni:

```
<meta property="og:title" content="A Basic HTML5 Template">  
<meta property="og:type" content="website">  
<meta property="og:url" content="https://www.sitepoint.com/a-basic-html5-template/">  
<meta property="og:description" content="A simple HTML5 Template for new projects.">  
<meta property="og:image" content="image.png">
```

Questi <meta> sfruttano il protocollo OpenGraph, ma ne esistono molti altri che puoi usare. Questi sono quelli che probabilmente utilizzerai più spesso. È possibile visualizzare un elenco completo delle opzioni meta OpenGraph disponibili sul sito Web Open Graph.

Quelli che ho incluso qui miglioreranno l'aspetto della pagina web quando è collegata a un post sui social media. Ad esempio, i cinque <meta> qui inclusi appariranno nelle schede sociali che incorporano i seguenti dati:

- un titolo per il contenuto
- il tipo di contenuto che viene consegnato
- l'URL canonico per il contenuto
- una descrizione del contenuto
- un'immagine da associare al contenuto

Quando vedi un post condiviso sui social media, vedrai spesso questi dati aggiunti automaticamente al post sui social media. Ad esempio, di seguito è riportato cosa apparirebbe in un tweet se includessi un collegamento alla home page di GitHub:



Favicon e icone touch

La sezione successiva del modello HTML5 include `<link>` che indicano le risorse da includere come favicon e icona AppleTouch:

```
<link rel="icon" href="/favicon.ico">  
<link rel="icon" href="/favicon.svg" type="image/svg+xml">  
<link rel="apple-touch-icon" href="/apple-touch-icon.png">
```

Il `favicon.ico` è per i browser “legacy”, il file deve essere incluso nella radice del tuo progetto, il browser lo troverà automaticamente. Il `favicon.svg` è per i browser moderni che supportano le icone SVG. L'ultimo elemento fa riferimento all'icona utilizzata sui dispositivi Apple quando la pagina viene aggiunta alla schermata iniziale dell'utente.

Includere un foglio di stile e script

Le ultime due parti significative del nostro modello HTML sono il riferimento a un foglio di stile e uno script. Entrambi sono opzionali, ovviamente:

```
<link rel="stylesheet" href="css/styles.css?v=1.0">
```

Il foglio di stile viene incluso utilizzando il tag `<link>` con un attributo appropriato. Un foglio di stile può essere incluso ovunque in un documento, ma di solito si trova all'interno

del tag <head>. A differenza delle versioni precedenti di HTML, non è necessario includere un attributo *type* (che non è mai stato necessario in primo luogo).

```
<script src="js/scripts.js"></script>
```

Il posizionamento <script> nella parte inferiore della pagina serve ai fini della velocità di caricamento della pagina. Quando un browser incontra uno script, interrompe il download e visualizza il resto della pagina mentre analizza lo script. Ciò fa sì che la pagina sembri caricarsi molto più lentamente quando vengono inclusi script di grandi dimensioni nella parte superiore della pagina prima di qualsiasi contenuto. Pertanto, la maggior parte degli script dovrebbe essere posizionata nella parte inferiore della pagina, in modo che vengano analizzati solo dopo che il resto della pagina è stato caricato. Ma tieni presente che in alcuni casi, potrebbe essere *necessario* posizionare lo script nell'intestazione del documento, perché desideri che abbia effetto prima che il browser inizi a visualizzare la pagina.

Poiché JavaScript è, per tutti gli scopi pratici, l'unico vero linguaggio di scripting utilizzato sul Web, e poiché tutti i browser presumono che tu stia utilizzando JavaScript anche quando non dichiari esplicitamente questo fatto, puoi tranquillamente tralasciare *type="text/javascript*, che spesso appare nel codice legacy.